PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-166977
(43)Date of publication of application : 22.06.2001

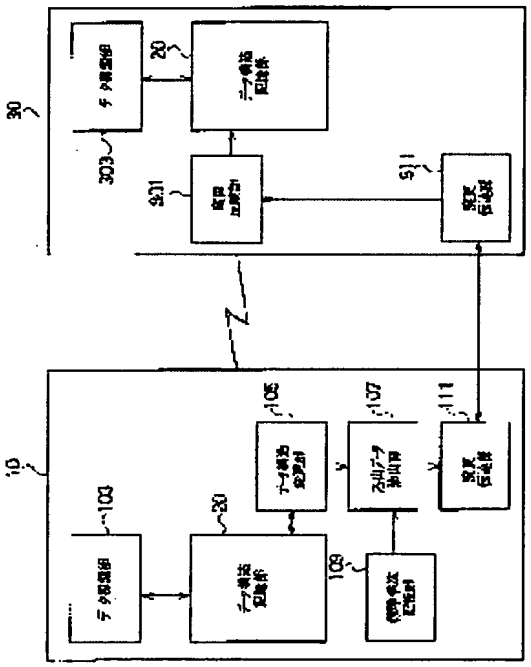
(51)Int.Cl. G06F 12/00
// G06F 17/30

(21)Application number : 11-353630
(22)Date of filing : 13.12.1999

(71)Applicant : TOSHIBA CORP
(72)Inventor : FUJIWARA MUTSUMI

(54) DEVICE, SYSTEM AND METHOD FOR MANAGING DATA STRUCTURE AND RECORDING MEDIUM HAVING DATA STRUCTURE MANAGEMENT PROGRAM

(57)Abstract:
PROBLEM TO BE SOLVED: To realize the simple and efficient maintenance management of a plurality of data structures arranged in one or more storage devices while reducing traffic.
SOLUTION: This data structure managing device 10 for managing a plurality of mutually equivalent data structures stored in one or more storage devices is provided with a data structure storing part 20 for storing an arbitrary first data structure, a data structure changing part 105 for changing the first data structure, a changed data extracting part 107 for extracting only the data necessary for making the first data structure at the origin of change equivalent to the second data structure at the destination of change among the changed data on the first data structure obtained by the data structure changing part 105, and a changed data transmitting part 111 for outputting the extracted changed data so that the data can be reflected on the second data structure.



LEGAL STATUS

[Date of request for examination] 25.09.2003

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開2001－166977
（P2001－166977A）
(43)公開日　平成13年６月22日(2001.6.22)

(51)Int.Cl.'	識別記号	Ｆ Ｉ	テーマコード'（参考）
Ｇ０６Ｆ　12/00	５３３	Ｇ０６Ｆ　12/00	５３３Ｊ　５Ｂ０７５
	５２０		５２０Ａ　５Ｂ０８２
// Ｇ０６Ｆ　17/30		15/419	３１０

審査請求　未請求　請求項の数20　ＯＬ　（全 22 頁）

(21)出願番号　特願平11－353630

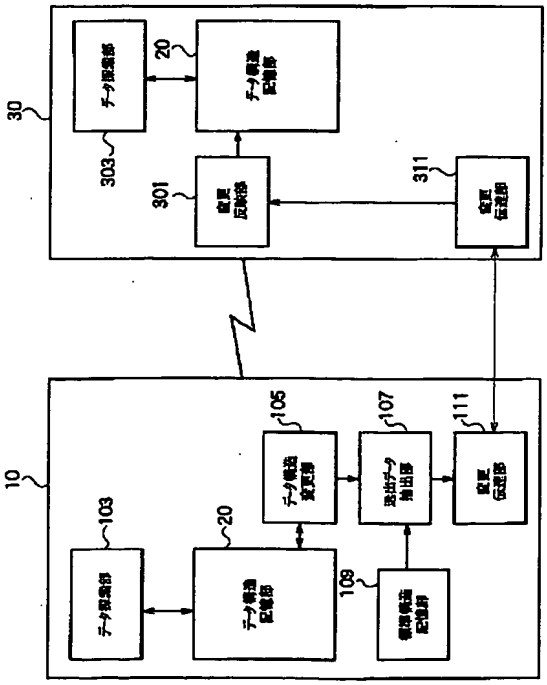
(22)出願日　平成11年12月13日(1999.12.13)

(71)出願人　000003078
株式会社東芝
神奈川県川崎市幸区堀川町72番地
(72)発明者　藤原　睦
神奈川県川崎市幸区柳町70番地　株式会社東芝柳町工場内
(74)代理人　100083806
弁理士　三好　秀和　（外７名）

Ｆターム(参考)　5B075 ND35 NK43 NK48 NR02 NR20
5B082 BA00 EA04 GB01 HA00 JA06

(54)【発明の名称】　データ構造管理装置、データ構造管理システム、データ構造管理方法およびデータ構造管理プログラムを格納する記録媒体

(57)【要約】
【課題】　１つまたは複数の記憶装置に配置された複数のデータ構造において、通信量を削減しつつ複数のデータ構造の容易かつ効率よい維持管理を実現する。
【解決手段】　１つまたは複数の記憶装置に格納され、相互に等価である複数のデータ構造を管理するデータ構造管理装置１０であって、任意の第１のデータ構造を格納するデータ構造記憶部２０と、第１のデータ構造を変更するデータ構造変更部１０５と、データ構造変更部１０５による第１のデータ構造上の変更データのうち、変更元の第１のデータ構造と、変更先の第２のデータ構造とを等価にするために必要なデータのみを抽出する変更データ抽出部１０７と、抽出された変更データを、前記第２のデータ構造に反映させるべく出力する変更データ送信部１１１とを具備する。



1

【特許請求の範囲】

【請求項１】　１つまたは複数の記憶装置に格納され、相互に等価である複数のデータ構造を管理するデータ構造管理装置であって、

任意の第１のデータ構造を格納するデータ構造記憶部と、

前記第１のデータ構造を変更するデータ構造変更部と、

前記データ構造変更部による前記第１のデータ構造上の変更データのうち、変更元の前記第１のデータ構造と、変更先の第２のデータ構造とを等価にするために必要なデータのみを抽出する変更データ抽出部と、

前記抽出された変更データを、前記第２のデータ構造に反映させるべく出力する変更データ送信部とを具備することを特徴とするデータ構造管理装置。

【請求項２】　前記変更データ抽出部は、

前記第１のデータ構造上のそれぞれのノードが有するポインタの変更情報の一部を、前記変更データとして抽出することを特徴とする請求項１に記載のデータ構造管理装置。

【請求項３】　前記変更データ抽出部は、

ノードが削除された際に、削除されたノードのポインタの非参照状態への変更を示すデータを前記変更データとして抽出することを特徴とする請求項２に記載のデータ構造管理装置。

【請求項４】　上記データ構造管理装置は、さらに、

前記第１のデータ構造と同一のソート順序でノードを配列する標準構造を格納する標準構造記憶部を具備し、

前記変更データ抽出部は、

前記標準構造記憶部に格納される前記標準構造において、前記データ構造変更部により行われた変更と同一の変更操作により変更されるポインタの変更情報を変更データとして抽出することを特徴とする請求項２または３に記載のデータ構造管理装置。

【請求項５】　前記第１のデータ構造は、ルートノードへのポインタを格納する記憶要素を含んで構成されることを特徴とする請求項１、２、３または４に記載のデータ構造管理装置。

【請求項６】　前記変更データ抽出部は、

前記変更データを、抽出されたポインタのアドレスを含むノードの先頭からのオフセットと、前記ノードの前記第１のデータ構造上の区間の値と、前記第１のデータ構造を指示する値との組として生成することを特徴とする請求項１、２、３、４または５に記載のデータ構造管理装置。

【請求項７】　上記データ構造管理装置は、さらに、

前記第２のデータ構造上でのノードの位置を探索する位置探索部を具備し、

前記変更データ反映部は、前記変更データに基づいて、前記位置探索部により探索された前記第２のデータ構造上のノードの位置における変更を行うことを特徴とする請求項６に記載のデータ構造管理装置。

2

【請求項８】　１つまたは複数の記憶装置に格納されるデータ構造を管理するデータ構造管理装置であって、

ルートノードへのポインタを格納する記憶要素を含んで構成される任意のデータ構造を格納するデータ構造記憶部と、

前記データ構造を変更するデータ構造変更部とを具備し、

前記記憶要素は、前記記憶要素を参照する変数へのポインタを格納することを特徴とするデータ構造管理装置。

【請求項９】　１つまたは複数の記憶装置に格納されるデータ構造を管理するデータ構造管理装置であって、

ルートノードへのポインタを格納する記憶要素を含んで構成される任意のデータ構造であって、前記データ構造上のノードの少なくとも１つは、前記記憶要素へのポインタを格納するデータ構造を格納するデータ構造記憶部と、

前記データ構造を変更するデータ構造変更部とを具備することを特徴とするデータ構造管理装置。

【請求項１０】　１つまたは複数の記憶装置に格納され、相互に等価である複数のデータ構造を管理するデータ構造管理装置であって、

任意の第１のデータ構造を格納するデータ構造記憶部と、

前記第１のデータ構造を変更するとともに、前記第１のデータ構造上の変更データのうち、変更元の前記第１のデータ構造と、変更先の第２のデータ構造とを等価にするために必要なデータのみを変更データとして抽出する変更データ抽出部と、

前記抽出された変更データを、前記第２のデータ構造に反映させるべく出力する変更データ送信部とを具備し、

前記変更データ抽出部は、前記第１のデータ構造上での変更を、予め定められた標準構造上での変更に変換することにより、前記変更データを算出することを特徴とするデータ構造管理装置。

【請求項１１】　前記変更データ抽出部は、

前記第１のデータ構造の一部を、前記標準構造の一部に変形することにより、前記変更データを算出することを特徴とする請求項１０に記載のデータ構造管理装置。

【請求項１２】　前記変更データ抽出部は、

前記第１のデータ構造上で、変更対象ノードおよび前記変更対象ノードの左隣および／または右隣のノードを順次ルートノードに移動して前記第１のデータ構造を変形することにより、前記変更データを算出することを特徴とする請求項１０または１１に記載のデータ構造管理装置。

【請求項１３】　１つまたは複数の記憶装置に格納され、相互に等価である複数のデータ構造を管理するデータ構造管理装置であって、

任意の第１のデータ構造と等価である第２のデータ構造

を格納するデータ構造記憶部と、
前記第１のデータ構造の変更データのうち、変更元の前記第１のデータ構造と、変更先の前記第２のデータ構造とを等価にするために必要なデータである変更データを入力する変更データ入力部と、
前記変更データを前記第２のデータ構造に反映する変更データ反映部とを具備することを特徴とするデータ構造管理装置。
【請求項１４】　１つまたは複数の記憶装置に格納され、相互に等価である複数のデータ構造を管理するデータ構造管理システムであって、
任意の第１のデータ構造を格納する第１のデータ構造記憶部と、
前記第１のデータ構造を変更するデータ構造変更部と、
前記データ構造変更部による前記第１のデータ構造上の変更データのうち、変更元の前記第１のデータ構造と、変更先の第２のデータ構造とを等価にするために必要なデータのみを抽出する変更データ抽出部と、
前記抽出された変更データを、前記第２のデータ構造に反映させるべく出力する変更データ送信部とを具備する第１のデータ構造管理装置と、
前記第１のデータ構造と等価である前記第２のデータ構造を格納する第２のデータ構造記憶部と、
前記第１のデータ構造管理装置の前記変更データ送信部から出力される前記変更データを入力する変更データ入力部と、
前記変更データを前記第２のデータ構造に反映する変更データ反映部とを具備する第２のデータ構造管理装置とを具備することを特徴とするデータ構造管理システム。
【請求項１５】　１つまたは複数の記憶装置に格納され、相互に等価である複数のデータ構造を管理するデータ構造管理方法であって、
任意の第１のデータ構造を変更するデータ構造変更ステップと、
前記データ構造変更ステップによる前記第１のデータ構造上の変更データのうち、変更元の前記第１のデータ構造と、変更先の第２のデータ構造とを等価にするために必要なデータのみを抽出する変更データ抽出ステップと、
前記抽出された変更データを、前記第２のデータ構造に反映させるべく出力する変更データ送信ステップとを含むことを特徴とするデータ構造管理方法。
【請求項１６】　前記変更データ抽出ステップは、
前記第１のデータ構造上のそれぞれのノードが有するポインタの変更情報の一部を、前記変更データとして抽出することを特徴とする請求項１５に記載のデータ構造管理方法。
【請求項１７】　上記データ構造管理方法は、さらに、
前記第１のデータ構造と同一のソート順序でノードを配列する標準構造を格納する標準構造格納ステップを含み、
前記変更データ抽出ステップは、
前記標準構造格納ステップにより格納される前記標準構造において、前記データ構造変更部により行われた変更と同一の変更操作により変更されるポインタの変更情報を変更データとして抽出することを特徴とする請求項１６に記載のデータ構造管理方法。
【請求項１８】　１つまたは複数の記憶装置に格納され、相互に等価である複数のデータ構造を管理するデータ構造管理処理をコンピュータに実行せしめるプログラムを格納するコンピュータ読み取り可能な記録媒体であって、
任意の第１のデータ構造を変更するデータ構造変更処理と、
前記データ構造変更処理による前記第１のデータ構造上の変更データのうち、変更元の前記第１のデータ構造と、変更先の第２のデータ構造とを等価にするために必要なデータのみを抽出する変更データ抽出処理と、
前記抽出された変更データを、前記第２のデータ構造に反映させるべく出力する変更データ送信処理とを含むことを特徴とするコンピュータ読み取り可能な記録媒体。
【請求項１９】　前記変更データ抽出処理は、
前記第１のデータ構造上のそれぞれのノードが有するポインタの変更情報の一部を、前記変更データとして抽出することを特徴とする請求項１８に記載のコンピュータ読み取り可能な記録媒体。
【請求項２０】　上記コンピュータ読み取り可能な記録媒体は、さらに、
前記第１のデータ構造と同一のソート順序でノードを配列する標準構造を格納する標準構造格納処理を含み、
前記変更データ抽出処理は、
前記標準構造格納処理により格納される前記標準構造において、前記データ構造変更部により行われた変更と同一の変更操作により変更されるポインタの変更情報を変更データとして抽出することを特徴とする請求項１９に記載のコンピュータ読み取り可能な記録媒体。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明は、データ構造管理装置、データ構造管理システム、データ構造管理方法およびデータ構造管理プログラムを格納する記録媒体に関する。特に、１つまたは複数の計算機の記憶装置に配置された複数のデータ構造において、あるデータ構造に対する変更を、当該データ構造と他のデータ構造との間の等価性維持に必要となる変更データのみを他のデータ構造に伝達して他のデータ構造に反映することにより、通信量を削減するとともに複数のデータ構造の容易かつ効率よい維持管理を実現するための技術に関する。
【０００２】
【従来の技術】従来より、計算機の主記憶を典型とする

１つまたは複数の番地付けされたランダムアクセス記憶装置（以下、「ＲＡＭ」と称する）の上に、ポインタを一定の手法で格納することによって種々のデータ構造が構築され、情報表現の効率化及び操作の高速化が図られてきた。

【０００３】これらデータ構造として、例えば、線形リスト、二分木、木、種々のグラフ等を、あるノードに対応づけられたＲＡＭの一部に、他のノードへのポインタを格納することによってデータ間の構造を表現する手法がある。これらのデータ構造は一定の操作および機能を提供するものである。

【０００４】しかしながら、１つまたは複数のＲＡＭ上に配置された複数のデータ構造が等価である機能を果たすように維持および管理する手法に関しては、これまで効率的な手法が知られていなかった。

【０００５】従来における複数のデータ構造の等価性維持の手法を、以下に説明する。

【０００６】第１に、データ構造の機能を変更する全ての更新操作およびこの更新操作に必要な探索操作を、１つのデータ構造に対してだけでなく、複数のデータ構造のどれに対しても全く同様の内容・順序で行うことにより、これら複数のデータ構造の機能的等価性を保証する手法がある。

【０００７】しかし、この第１の手法においては、更新（変更）を反映すべき他のデータ構造（以下、「コピー側データ構造」と称する）を維持および管理するのに必要な手間すなわち計算時間は、変更元のデータ構造（以下、「マスター側データ構造」と称する）を維持および管理するのに必要な手間と同一である。

【０００８】具体的には、データ構造を変更するためには、多くの場合、この変更に先だって複雑な探索が実行され、この探索結果に依存して変更の内容が異なる。従って、第１の手法では、コピー側のデータ構造でもマスター側のデータ構造と同様に、必要な探索を含む変更手続きを実行しなければ双方のデータ構造の等価性を実現することができなかった。しかし、この第１の手法では、マスター側のデータ構造への変更処理とコピー側のデータ構造への変更反映処理とは、同等の計算時間（リソース）を必要とする。このため、他の処理、例えば探索のみの要求等を処理できる余地は、双方での大差がないことになる。従って、データ構造の等価性の実現に多大な計算時間を要し、例えばコピー側ではマスター側と異なった多くの探索要求を処理し、一方マスター側は変更要求の処理に専念する、等の分散環境における効率的な処理の分担が実現できなかった。すなわち、この第１の手法では、１つのデータ構造に加えられた変更のみを効率的に他のデータ構造に反映させて、他のデータ構造の管理コストを下げることは望めなかった。

【０００９】第２に、ポインタの値も含めてデータ構造をＲＡＭの内容として全く同一に保つために、変更操作

において変更されたポインタをすべて伝送して変更先に逐一反映するという手法がある。この第２の手法は、それぞれのデータ構造（より正確にはそれぞれのデータ構造を構成する各ノード）を、ＲＡＭ上の同一の番地とみなせる部分に格納し、１つのデータ構造への更新（すなわち、ポインタの変更情報）を各番地の記憶内容の変更として他のデータ構造を表わすＲＡＭにコピーする。１つの記憶装置（あるいは計算機）上でも複数の記憶装置（あるいは計算機）間でも、適切なアドレス変換を行なう手段を併用すれば、複数の等価なデータ構造を同一の番地に同一表現のポインタを格納することで構築することができる。複数のデータ構造をこのように構築し、ポインタも他のデータ（コード）と同様に単純にコピーすることによって、データ構造の変更を他のデータ構造に反映することができる。

【００１０】しかし、この第２の手法は、機能的には等価であるが、ポインタ構造としては異なる形態をとることを許す複数のデータ構造に対しては適用することができない。例えば、アクセス頻度の高いノードをルートのノードとすべく、ポインタ構造を適宜調整する自己調整型データ構造（self-adjusting data structure）と呼ばれるデータ構造がある。この自己調整型データ構造は、機能の変更を伴わない探索操作においても、データ構造の形状を、機能的に等価な他の形状に変えてしまう。この自己調整型データ構造において、格納されるポインタの値を同一にして等価性を維持するためには、探索も含めた全ての操作によって発生するポインタ（各番地の内容）の変更を全てコピーしなければならない。このコピーのためには、膨大な量のポインタ変更情報を他のデータ構造（あるいは他のデータ構造の管理手段）に伝達しなければならず、管理の手間だけでなく通信量も非常に多くなる。

【００１１】あるいは、ハイ・バランスト・ツリー（hight-balanced-tree）のようなデータ構造の場合には、１回の変更操作で多数のポインタが変更されることがあり、このため、伝送すべき変更データの量が多くなってしまう。また、第２の手法においては、マスター側とコピー側とでデータ構造の構成（形態）が全く一致していなければならないという制約があった。すなわち、例えば二分木のデータ構造の場合、二分探索木（バイナリー・サーチ・ツリー：binary search tree）として等価というだけでなく、対応する同じノード間の親子関係が全て一致していなければならない。（もっとも、上記のアドレス変換が適切に行なわれれば、各ノードのアドレスおよびポインタ値まで同一である必要はない。）換言すれば、この第２の手法では、マスター側とコピー側で、バイナリー・サーチ・ツリーとしては等価であるが具体的なバイナリー・ツリーの構成が異なるデータ構造を使用することができない。

【００１２】このため、例えば、マスター側とコピー側

とでツリーの構成を全く別の手法で管理する場合、すなわち、例えばマスター側は上記のハイ・バランスト・ツリー（hight-balanced-tree）を採用してツリーの構成を管理し、一方コピー側はスプレー・ツリー（splay tree）を採用して、探索操作が行なわれた場合にもツリーの構成を変更するような場合には、マスター側で変更を加える前のデータ構造の構成（すなわち、ツリーの形状）とコピー側でその変更データを反映する前のデータ構造の構成がそもそも異なる。したがって単純にマスター側のデータ構造を変更した際、変更されたポインタをコピー側のデータ構造に反映することはできない。

【００１３】特に、物理的に異なる計算機の主記憶上にそれぞれ配置された複数のデータ構造の等価性を、上記の第１あるいは第２の手法により実現しようとすれば、これら計算機間を接続する通信路（ネットワーク）に多大な負荷をかけることになり、通信路を含むシステム全体に重大な影響を及ぼす。

【００１４】さらに、例えば、更新操作の場合にのみデータ構造の形状を変更するデータ構造である各種の平衡木等においても、形状の変更に伴って多数のポインタが変更される。このため、これらのポインタの変更情報を全て伝達するためには、データ構造を格納したＲＡＭの間を接続する通信路には、上記の自己調整型データ構造ほどではないにしろ、多大な負荷がかかることになる。

【００１５】

【発明が解決しようとする課題】以上説明したように、従来のデータ構造管理手法には、１つまたは複数のＲＡＭ上に配置された複数のデータ構造が等価である機能を果たすように維持および管理することは、多くの計算時間を要するとともに通信量が多く、非効率的であったという問題点を解決するためになされたものである。

【００１６】そして、その目的とするところは、１つまたは複数のＲＡＭ（記憶領域）上に配置されたデータ構造への変更を、少ない通信量および計算時間で、他のデータ構造に反映することによって、効率よく複数のデータ構造の等価性を維持、管理することのできるデータ構造管理装置、データ構造管理システム、データ構造管理方法およびデータ構造管理プログラムを格納する記録媒体を提供する点にある。

【００１７】また、本発明の他の目的は、相互に異なるアドレスに配置されるデータ構造の一部分または全体の間での等価性を容易に維持、管理することにある。

【００１８】

【課題を解決するための手段】上記の課題を解決するための本発明の特徴は、あるデータ構造に対するポインタの変更データのうち、所定の標準構造に基づいて、複数のデータ構造の間の機能的な等価性を維持するために必要となる変更データを抽出し、この抽出された変更データのみを変更を反映すべきデータ構造に送出する点にある。

【００１９】かかる機能を実現するための、本発明の第１の特徴は、１つまたは複数の記憶装置に格納され、相互に等価である複数のデータ構造を管理するデータ構造管理装置であって、任意の第１のデータ構造を格納するデータ構造記憶部と、前記第１のデータ構造を変更するデータ構造変更部と、前記データ構造変更部による前記第１のデータ構造上の変更データのうち、変更元の前記第１のデータ構造と、変更先の第２のデータ構造とを等価にするために必要なデータのみを抽出する変更データ抽出部と、前記抽出された変更データを、前記第２のデータ構造に反映させるべく出力する変更データ送信部とを具備することを特徴とするデータ構造管理装置を提供する点にある。

【００２０】本発明の第２の特徴は、前記変更データ抽出部は、前記第１のデータ構造上のそれぞれのノードが有するポインタの変更情報の一部を、前記変更データとして抽出する点にある。

【００２１】本発明の第３の特徴は、前記変更データ抽出部は、ノードが削除された際に、削除されたノードのポインタの非参照状態への変更を示すデータを前記変更データとして抽出する点にある。

【００２２】本発明の第４の特徴は、上記データ構造管理装置は、さらに、前記第１のデータ構造と同一のソート順序でノードを配列する標準構造を格納する標準構造記憶部を具備し、前記変更データ抽出部は、前記標準構造記憶部に格納される前記標準構造において、前記データ構造変更部により行われた変更と同一の変更操作により変更されるポインタの変更情報を変更データとして抽出する点にある。

【００２３】本発明の第５の特徴は、前記第１のデータ構造は、ルートノードへのポインタを格納する記憶要素を含んで構成される点にある。

【００２４】本発明の第６の特徴は、前記変更データ抽出部は、前記変更データを、抽出されたポインタのアドレスを含むノードの先頭からのオフセットと、前記ノードの前記第１のデータ構造上の区間の値と、前記第１のデータ構造を指示する値との組として生成する点にある。

【００２５】本発明の第７の特徴は、上記データ構造管理装置は、さらに、前記第２のデータ構造上でのノードの位置を探索する位置探索部を具備し、前記変更データ反映部は、前記変更データに基づいて、前記位置探索部により探索された前記第２のデータ構造上のノードの位置における変更を行う点にある。

【００２６】本発明の第８の特徴は、１つまたは複数の記憶装置に格納されるデータ構造を管理するデータ構造管理装置であって、ルートノードへのポインタを格納する記憶要素を含んで構成される任意のデータ構造を格納するデータ構造記憶部と、前記データ構造を変更するデータ構造変更部とを具備し、前記記憶要素は、前記記憶

要素を参照する変数へのポインタを格納することを特徴とするデータ構造管理装置を提供する点にある。

【００２７】本発明の第９の特徴は、１つまたは複数の記憶装置に格納されるデータ構造を管理するデータ構造管理装置であって、ルートノードへのポインタを格納する記憶要素を含んで構成される任意のデータ構造であって、前記データ構造上のノードの少なくとも１つは、前記記憶要素へのポインタを格納するデータ構造を格納するデータ構造記憶部と、前記データ構造を変更するデータ構造変更部とを具備することを特徴とするデータ構造管理装置を提供する点にある。

【００２８】本発明の第１０の特徴は、１つまたは複数の記憶装置に格納され、相互に等価である複数のデータ構造を管理するデータ構造管理装置であって、任意の第１のデータ構造を格納するデータ構造記憶部と、前記第１のデータ構造を変更するとともに、前記第１のデータ構造上の変更データのうち、変更元の前記第１のデータ構造と、変更先の第２のデータ構造とを等価にするために必要なデータのみを変更データとして抽出する変更データ抽出部と、前記抽出された変更データを、前記第２のデータ構造に反映させるべく出力する変更データ送信部とを具備し、前記変更データ抽出部は、前記第１のデータ構造上での変更を、予め定められた標準構造上での変更に変換することにより、前記変更データを算出することを特徴とするデータ構造管理装置を提供する点にある。

【００２９】本発明の第１１の特徴は、前記変更データ抽出部は、前記第１のデータ構造の一部を、前記標準構造の一部に変形することにより、前記変更データを算出する点にある。

【００３０】本発明の第１２の特徴は、前記変更データ抽出部は、前記第１のデータ構造上で、変更対象ノードおよび前記変更対象ノードの左隣および／または右隣のノードを順次ルートノードに移動して前記第１のデータ構造を変形することにより、前記変更データを算出する点にある。

【００３１】本発明の第１３の特徴は、１つまたは複数の記憶装置に格納され、相互に等価である複数のデータ構造を管理するデータ構造管理装置であって、任意の第１のデータ構造と等価である第２のデータ構造を格納するデータ構造記憶部と、前記第１のデータ構造の変更データのうち、変更元の前記第１のデータ構造と、変更先の前記第２のデータ構造とを等価にするために必要なデータである変更データを入力する変更データ入力部と、前記変更データを前記第２のデータ構造に反映する変更データ反映部とを具備することを特徴とするデータ構造管理装置を提供する点にある。

【００３２】本発明の第１４の特徴は、１つまたは複数の記憶装置に格納され、相互に等価である複数のデータ構造を管理するデータ構造管理システムであって、任意の第１のデータ構造を格納する第１のデータ構造記憶部と、前記第１のデータ構造を変更するデータ構造変更部と、前記データ構造変更部による前記第１のデータ構造上の変更データのうち、変更元の前記第１のデータ構造と、変更先の第２のデータ構造とを等価にするために必要なデータのみを抽出する変更データ抽出部と、前記抽出された変更データを、前記第２のデータ構造に反映させるべく出力する変更データ送信部とを具備する第１のデータ構造管理装置と、前記第１のデータ構造と等価である前記第２のデータ構造を格納する第２のデータ構造記憶部と、前記第１のデータ構造管理装置の前記変更データ送信部から出力される前記変更データを入力する変更データ入力部と、前記変更データを前記第２のデータ構造に反映する変更データ反映部とを具備する第２のデータ構造管理装置とを具備することを特徴とするデータ構造管理システムを提供する点にある。

【００３３】本発明の第１５の特徴は、１つまたは複数の記憶装置に格納され、相互に等価である複数のデータ構造を管理するデータ構造管理方法であって、任意の第１のデータ構造を変更するデータ構造変更ステップと、前記データ構造変更ステップによる前記第１のデータ構造上の変更データのうち、変更元の前記第１のデータ構造と、変更先の第２のデータ構造とを等価にするために必要なデータのみを抽出する変更データ抽出ステップと、前記抽出された変更データを、前記第２のデータ構造に反映させるべく出力する変更データ送信ステップとを含むことを特徴とするデータ構造管理方法を提供する点にある。

【００３４】本発明の第１６の特徴は、前記変更データ抽出ステップは、前記第１のデータ構造上のそれぞれのノードが有するポインタの変更情報の一部を、前記変更データとして抽出する点にある。

【００３５】本発明の第１７の特徴は、上記データ構造管理方法は、さらに、前記第１のデータ構造と同一のソート順序でノードを配列する標準構造を格納する標準構造格納ステップを含み、前記変更データ抽出ステップは、前記標準構造格納ステップにより格納される前記標準構造において、前記データ構造変更部により行われた変更と同一の変更操作により変更されるポインタの変更情報を変更データとして抽出する点にある。

【００３６】本発明の第１８の特徴は、１つまたは複数の記憶装置に格納され、相互に等価である複数のデータ構造を管理するデータ構造管理処理をコンピュータに実行せしめるプログラムを格納するコンピュータ読み取り可能な記録媒体であって、任意の第１のデータ構造を変更するデータ構造変更処理と、前記データ構造変更処理による前記第１のデータ構造上の変更データのうち、変更元の前記第１のデータ構造と、変更先の第２のデータ構造とを等価にするために必要なデータのみを抽出する変更データ抽出処理と、前記抽出された変更データを、

前記第２のデータ構造に反映させるべく出力する変更データ送信処理とを含むことを特徴とするコンピュータ読み取り可能な記録媒体を提供する点にある。

【００３７】本発明の第１９の特徴は、前記変更データ抽出処理は、前記第１のデータ構造上のそれぞれのノードが有するポインタの変更情報の一部を、前記変更データとして抽出する点にある。

【００３８】本発明の第２０の特徴は、上記コンピュータ読み取り可能な記録媒体は、さらに、前記第１のデータ構造と同一のソート順序でノードを配列する標準構造を格納する標準構造格納処理を含み、前記変更データ抽出処理は、前記標準構造格納処理により格納される前記標準構造において、前記データ構造変更部により行われた変更と同一の変更操作により変更されるポインタの変更情報を変更データとして抽出する点にある。

【００３９】

【発明の実施の形態】以下、図面を参照して、本発明に係るデータ構造管理装置、データ構造管理システム、データ構造管理方法およびデータ構造管理プログラムを格納する記録媒体の実施の形態を、詳細に説明する。

【００４０】第１の実施形態

以下、図１から図２０を参照して、本発明に係るデータ構造管理装置、データ構造管理システム、データ構造管理方法およびデータ構造管理プログラムを格納する記録媒体の第１の実施形態を詳細に説明する。第１の実施形態は、バイナリー・サーチ・ツリー（binary search tree）のデータ構造を複数のＲＡＭ上に構築し、これらデータ構造の間で効率よく機能的な等価性を維持する機能を提供する。

【００４１】図１は、第１の実施形態に係るデータ構造管理システムの一般的な機能構成を示すブロック図である。図１に示すように、第１の実施形態に係るマスター側のデータ構造管理装置１０は、データ構造記憶部２０と、データ探索部１０３と、データ構造変更部１０５と、送出データ抽出部１０７と、標準構造記憶部１０９と、変更送信部１１１とにより構成される。第１の実施形態に係るコピー側のデータ構造管理装置３０は、データ構造記憶部２０と、データ探索部３０３と、変更反映部３０１と、変更受信部３１１とにより構成される。

尚、図１の構成は、データ構造管理装置１０のデータ構造記憶部２０をマスターとし、このマスターにおけるデータ構造の変更を、コピー側のデータ構造管理装置３０のデータ構造記憶部２０に反映させるマスター・スレーブ型の構成の一例である。第１の実施形態は、この他、例えばデータ構造管理装置１０およびデータ構造管理装置３０の構成を併せ持つ複数のデータ構造管理装置を相互に接続して、相互にデータ構造変更データを交換するピア・ツー・ピア型の構成とするなど、任意に構成することができる。また、図１の構成は、変更を反映すべきコピー側のデータ構造を１つとして説明しているが、通信機能が複数の相手に同一の変更データを送信できればコピー側のデータ構造を複数としてもよい。

【００４２】第１の実施形態においては、変更送信部１１１および変更受信部３１１により提供される通信機能により、伝送された変更データを変更反映部３０１がコピー側のデータ構造記憶部２０のデータ構造に反映する。この変更データがコピー側データ構造２０に反映された後には、マスター側データ構造２０のデータ探索部１０３とコピーデータ構造のデータ探索部３０３とは全く同等に機能する。すなわち、双方のデータ構造２０は、同じ探索要求に対して同じ結果を返すことができる。従って、双方のデータ構造の等価性が実現されることによって、マスター・コピー間でデータが共有できる。

【００４３】マスター側およびコピー側のデータ構造記憶部２０のデータ構造は、相互に機能的に等価な任意のデータ構造である。ここで、機能的に等価とは、各ノードの論理的な順序（本実施形態の場合はソート順）を同じくすること、すなわち機能的に同値であることをいい、相互のポインタ構造自体は異なっていてもよい。

【００４４】データ探索部１０３、３０３は、アプリケーションからの要求に従い、データ構造記憶部２０にそれぞれ格納されるデータ構造中のデータを探索する。

【００４５】マスター側データ構造管理装置１０中のデータ構造変更部１０５は、データ構造記憶部２０ａ中に格納されるマスター側データ構造を適宜変更し、変更されたポインタ情報である変更データを、送出データ抽出部１０７に通知する。

【００４６】送出データ抽出部１０７は、変更された変更データのうち、標準構造記憶部１０９に格納される標準となるデータ構造に照らして、データ構造の各ノードの論理的順番を変化させるポインタの情報のみを抽出し、この抽出されたポインタ情報を変更送信部１１１に送出する。

【００４７】変更送信部１１１は、抽出されたポインタ情報を、マスター側データ構造管理装置１０と同一ＲＡＭ内、同一計算機内あるいはローカルまたはリモートに接続される他の計算機にある変更受信部３１１に送信する。

【００４８】一方、コピー側のデータ構造管理装置３０中の変更受信部３１１は、マスター側データ構造管理装置１０の変更送信部１１１から受信したポインタ情報（変更データ）を、変更反映部３０１に送出する。

【００４９】変更反映部３０１は、変更受信部３１１から受信したポインタ情報（変更データ）を、データ構造記憶部２０に格納されるコピー側データ構造に反映する。

【００５０】次に、第１の実施形態に係るデータ構造管理装置のハードウエア構成を説明する。本発明に係るデータ構造管理システムにおけるデータ構造管理装置１０

およびデータ構造管理装置３０は、いわゆる汎用機、ワークステーション、ＰＣ、ネットワーク端末等の各種コンピュータ単体あるいは各コンピュータを相互接続したシステムに実装される。本発明の実施形態で用いる各コンピュータシステムのハードウエアは、各種処理を行うためのＣＰＵと、プログラムメモリ・データメモリ等のメモリと、ＦＤ・ＣＤ等の外部記憶装置と、キーボード・マウス等の入力装置と、ディスプレー・プリンタ等の出力装置とを備える。

【００５１】尚、本発明に係るデータ管理の各種処理を実現するためのプログラムは、各種記録媒体に保存することができる。かかる記録媒体を、上記ハードウエアを具備するコンピュータにより読み出し、当該プログラムを、主記憶等の記憶装置に格納されるデータ構造にアクセスして実行することにより、本発明が実施される。ここで、記録媒体とは、例えば、メモリ・磁気ディスク・光ディスク等、プログラムを記録することができる装置全般を含む。

【００５２】次に、共用すべき複数のデータ構造を、ＲＡＭ上にバイナリー・サーチ・ツリーで構成する場合を例として、第１の実施形態に係るデータ構造管理システムの処理を説明する。

【００５３】図２は、本発明の第１の実施形態に係るデータ構造管理システムの行う、データ構造の変更および他のデータ構造への変更の反映の処理の概略を示すフローチャートである。まず、データ構造変更部１０５は、マスター側のデータ構造記憶部２０中に格納されるデータ構造を変更する（ステップＳ１０）。送出データ抽出部１０７は、このデータ構造における変更されたポインタ情報のうち、標準構造記憶部１０９に格納される標準構造上での構造を変化させるポインタ情報を抽出する（ステップＳ２０）。変更送信部１１１は、この抽出されたポインタ情報を、コピー側データ構造管理装置３０の変更受信部３１１に送信する（ステップＳ３０）。コピー側データ構造管理装置３０では、変更受信部３１１により受信された変更データ（ポインタ情報）を変更反映部３０１に送出する。変更反映部３０１は、データ構造記憶部２０に格納されるデータ構造にアクセスし、受信された変更データ（ポインタ情報）によりコピー側データ構造を更新する。

【００５４】まず、第１の実施形態が用いるデータ構造の一例として、１つのノードにつき２つのポインタを格納する表現手法を説明する。ここで、ポインタとは、他のノードの番地を意味する。

【００５５】バイナリー・ツリーは、１つのノードにつき２つのポインタを使用するだけで、親から子および子から親への双方向のリンクを実現することができる。

【００５６】図３は、この双方向のリンクを実現する、leftmost-child-right-sibling representationと呼ばれる２ポインタ表現手法を説明する。

【００５７】図４は、図３の２ポインタ表現手法を各ノードをツリー状に接続した二分木（バイナリー・ツリー）で表現した様子を示す。

【００５８】図３において、各ノードの２つのポインタは、状況に応じて次のようなノードをポイントする。

【００５９】（１）一方（右）のポインタ
左の子があれば左の子を、左の子がなくかつ右の子があれば右の子を指す。

【００６０】子が全くなければnil（空ポインタ）である。

【００６１】（２）他方（左）のポインタ
自分が左の子でかつ右の兄弟（自分の親の右の子）があれば右の兄弟を、それ以外で親がある、すなわちルートノードでない場合は親を指す。

【００６２】親がない、すなわちルートノードである場合はnil（空ポインタ）である。

【００６３】尚、図３のように適切に配置された図上で一覧すると見過ごしがちであるが、あるノードが左右の子のどちらか一方しかない場合には、子を指す（右の）ポインタが左右どちらの子を指しているのかはポインタを見ただけでは判断できない。この子の左右を区別するためには、各ノードについて１ビットの余分な情報が必要である。この１ビットの情報の内容は、各ノード自身が左右どちらの子かを示してもよく、一方、各ノードの子へのポインタが左右でどちらの子かを示してもよい。バイナリー・サーチ・ツリー（二分探索木）の場合、キーの比較を行って左右どちらの子かを決定するということも可能である。但し、同一のキーを複数のノードが持ちうるような場合にはこの方法は使えない。一般的に、バイナリー・ツリーの構造を作るためには、この１ビットの情報が必要である。

【００６４】一方、兄弟または親を指すもう１つの（左の）ポインタに関しては、それが右の兄弟を指しているのか親を指しているのかは、最高でも２回ポインタをたどれば判定できる。すなわち、右のポインタが右の兄弟を指すのは、右のポインタが指すノードの右のポインタが指すノードの左のポインタが当該ノードに一致する場合に限られるのである。但し、このポインタから、そのノードがルートか否かの情報も得る必要がある。上記の例ではこれを空ポインタか否かで判定可能である。

【００６５】上記のように、バイナリー・ツリーは、１ノードにつき２つのポインタと１ビットの情報を格納することによって実現できる。この構造によって、３つのポインタを使用する一般に知られる構造と同じく、親から左の子を決定する操作、親から右の子を決定する操作、及び子から親を決定する操作が、それぞれ所定時間以内で実行できる。但し、１つの操作につき２つ以上のポインタ及びビットを参照するので、これらの操作の実行時間は３つのポインタを使用する場合より長い。逆に、各ノードの内容を設定したり変更したりする操作に

おいては、ポインタの数が少ない分実行時間が短縮される効果がある。

【００６６】次に、第１の実施形態で使用される２ポインタ表現のノード構成およびツリーへの実装を具体的に説明する。

【００６７】まず、第１の実施形態に係るデータ構造の、記憶装置（ＲＡＭ）への実装手法およびノードのポインタのフォーマットを説明する。

【００６８】使用される記憶装置は、番地付けされた記憶要素で構成されており、各記憶要素には任意の番地を格納できるものとする。すなわち、番地がポインタとして使用される。以下において、番地mの要素に言及する場合は、［m］で表す。

【００６９】図５に示すように、１つのノードのポインタ部分には、偶数番地の要素［2n］とそれに続く要素［2n+1］が充てられる。ノードを指示する場合には、そのポインタ部分の偶数番地を用いる。すなわち、［2n］および［2n+1］が充てられたノードは、「ノード 2n」ということになる。

【００７０】これらのノードを使用して構成されるバイナリー・ツリーのルートへのポインタを別の要素に格納し、この別の要素も含めた全体を一つのデータ構造として扱う。このルートへのポインタを格納する別の記憶要素を、当該ツリーのポット（鉢）と呼ぶ。ルートノードには、親または右の兄弟へのポインタの代わりにポットへのポインタを格納する。図５に示すように、ポットの番地をm、ルートノードを2nとすると、［m］には2nが、［2n］にはmが設定される。一方、ポット mに対応するツリーが空のときは［m］には０（nil）が設定される。このように、ノードの集まりであるバイナリー・ツリーに一意に対応するポットを導入し、これらを一体として扱う点は、第１の実施形態の特徴の１つである。

【００７１】なお、番地０はnilポインタとして用いられるので、［0］はノードとしては使用しない。但し、ポットとして使用することはできる。nilポインタには記憶要素が対応づけられていない番地、またはノードとして使用しない番地と判定できる番地ならいずれの番地を用いてもよい。

【００７２】図５に示すように、ルート以外のノード 2nが左の子でかつ右の兄弟2rがある場合、［2n］には2r+1が設定される。それ以外の場合、すなわち2nが左の子でかつ右の兄弟がないか、または2nが右の子である場合には、［2n］には親ノード２pの番地2pが設定される。前者の場合、［2n］の値は奇数（すなわち、ＬＳＢが１）なので、偶数（すなわち、ＬＳＢが０）である後者の場合とＬＳＢ（LeastSignificand Bit)の１ビットのみで区別することができる。この区別は必須ではないが、手続の簡素化と性能の向上には有効である。

【００７３】一方、図５に示すように、各ノード 2nのもう１つのポインタ［2n+1］には、そのノードに子がない場合は０（nil）が設定される。2nに左の子2qがある場合には、右の子の有無に関係なく［2n+1］には2qが設定される。2nに左の子がなく、右の子2rのみがある場合は、［2n+1］には2r+1が設定される。前者の場合、［2n+1］の値は偶数（ＬＳＢが０）なので、奇数（ＬＳＢが１）である後者の場合とＬＳＢの１ビットのみで区別することができる。

【００７４】上記のポット、ルート、親ノード、子ノードの関係は、以下の６つの場合がある。

【００７５】

a.ツリーが空の場合のポットのみ（図６参照）
b.対応するツリーを有するポットの場合（図７参照）
c.子がないノードの場合（図８参照）
d.右の子のみを持つノードの場合（図９参照）
e.左の子のみを持つノードの場合（図１０参照）
f.左右両方の子を持つノードの場合（図１１参照）

上記の６つの場合に基づいて、ノード 2xからその親子兄弟のノードに関する接続情報は、例えば以下のようにして判定できる。尚、コメント中に、各判定条件について、2xが上記a.からf.のどの場合のどのノードに対応するかを示す。

【００７６】

```
（１）ノード2xの親、兄弟の判定
if
  ([[2x]]＝2x) then  2xはroot、potは[2x]  /* b.ノード2n   */
else if
  (([2x]&1)＝1) then  2xは左の子、親は[[2x]-1]、右の兄弟は[2x]-1
                                        /*  f.ノード2q   */
else if
  (([[2x]+1]&1)＝1) then 2xは右の子、親は[2x]、左の兄弟はない
                                        /*  d.ノード2ｒ   */
else
                2xは右の子、親は[2x]、左の兄弟は[[2x]+1]
                                        /*  f.ノード2r    */
（２）ノード 2xの子の判定
if
```

```
  ([2x+1]＝0)  then 2xに子はない。          /* c.ノード2p     */
else if
  (([2x+1]&1)＝1) then 2xの左の子はなく、右の子は[2x+1]-1
                                          /* d.ノード2p     */
else if
  ((([2x+1]]&1)＝1)ならば2xの左の子は[2x+1]、右の子は[[2x+1]]-1
                                          /* f.ノード2p     */
else
                    2xの左の子は[2x+1]、右の子はない
                                          /* e.ノード2p     */
```

さらに、上記のように構成されたバイナリー・ツリーを変更する手続きには、次のような手続きが可能である。

【００７７】（１）空のバイナリー・ツリーに対応するポット mに、ノード 2nをルートとして対応付ける。すなわち、ノード 2nをルートとするバイナリー・ツリーを対応づける。この場合、上記のa.からb.へ構成が変更される。［2n］にポインタmを設定し、［m］にポインタ2nを設定する。

【００７８】（２）右の子がないノード 2pに、ノード 2rを右の子として付加する。すなわち、ノード 2rをルートとするバイナリー・ツリーを付加する。右の子がない状態が、上記のc.およびe.のいずれであるかは、上記の判定手続きで識別できる。

【００７９】c.からd.への構成の変更の場合、［2r］にポインタ2pを設定し、［2p+1］にポインタ2r+1を設定する。

【００８０】一方、e.からf.への構成の変更の場合、［2r］にポインタ2pを設定し、[[2p+1]]にポインタ2r+1を設定する。

【００８１】（３）左の子がないノード 2pに、ノード 2qを左の子として付加する。すなわち、ノード 2qをルートとするバイナリー・ツリーを付加する。左の子がない状態が上記のc.およびe.のいずれであるかは上記の判定手続きで識別できる。

【００８２】c.からe.への構成の変更の場合、［2q］にポインタ2pを設定し、［2p+1］にポインタ2qを設定する。

【００８３】一方、d.からf.への構成の変更の場合、［2q］に［2p+1］を設定し、［2p+1］にポインタ2qを設定する。

【００８４】（４）ノード 2xあるいはノード2xをルートとするサブツリーを、その親ノードから分離する。ノード2xがd.e.f.の2q、2rのいずれに該当するかは上記の判定手続きで識別できる。

【００８５】d.の2rの場合、［［2x］+1］に０（nil）を設定する。

【００８６】e.の2qの場合、［［2x］+1］に０（nil）を設定する。

【００８７】f.の2rの場合、［［［2x］+1］］に［2x］を設定する。

【００８８】f.の2qの場合、［［［2x］-1］+1］に［2x］を設定する。

【００８９】図１２は、図３および図４のバイナリーツリーを第１の実施形態の手法で扱ったデータ構造を、ポットも含めて正確に図示した図である。尚、各ノードの先頭番地は偶数番地である。

【００９０】図１３は、図１２のデータ構造と、バイナリー・サーチ・ツリーとして等価な構成である。このデータ構造は、どのノードも左の子を持たない。すなわち、どのノードも右の子のみを持つかあるいは子を持たないノードである。このデータ構造は、ルートノードａから末端のノードｊまで昇順または降順のキーでソートされている。尚、ポットおよびノードの番地は、図１２のデータ構造と同一である。この図１３の構成のデータ構造を標準構造と称する。この標準構造は、標準構造記憶部１０９に格納され、送出データ抽出部１０７により参照される。

【００９１】次に、図１２のデータ構造に基づいて、第１の実施形態に係るデータ構造管理の処理における手順を具体的に説明する。

【００９２】第１の実施形態に係るデータ構造管理装置は、上記のバイナリー・サーチ・ツリーの２ポインタ表現による複数のデータ構造を、等価であるデータの複数の表現として管理し、これらの１つに加えられた変更を、他のデータ構造に逐次反映することによって、これらを相互に同一（あるいは等価）のバイナリー・サーチ・ツリーとして機能させる。

【００９３】以下では、第１の実施形態に係るデータ構造管理システムが、例えば２台の計算機のＲＡＭ上の同一の番地（アドレス）に、各ノードおよびポットを配置した２つのデータ構造を管理する場合について説明する。

【００９４】（１）ノードの削除の場合

まず、２つの計算機上のデータ構造が、いずれも図１２と全く同じであるとする。ここで、変更操作として一方において５０をキーとするルートノードを削除し、６０をキーとするノードを新たなルートノードとする構成を作るものとする。図１４は、この変更操作後のデータ構造を示す。図１４中において、図１２と異なる、すなわち変更されたポインタは破線矢印で示す。削除されたノ

ードｆの２つのポインタは別として、削除後のバイナリー・サーチ・ツリーにおいて、ポットも含めて４つのポインタが変更されていることがわかる。

【００９５】ここで、第１の実施形態に係るデータ構造管理装置１０の変更送信部１１１は、これら変更されたポインタ及びその格納番地のすべてを、逐一他方のデータ構造を管理するデータ構造管理装置の変更伝達部３１１に送信するのではなく、図１３の標準構成に対して同一の変更操作を行なった場合に変更されるポインタのうち、ノードの先頭アドレスである偶数番地に格納されたポインタ情報だけを番地とともに送信する。

【００９６】図１５は、図１４に対応する標準構造を示す。この図１の標準構造において、図１３と異なっている４つのポインタ値、すなわち、ｇ番地の値ｅ、ｅ＋１番地の値ｇ＋１、ｆ番地の値０、ｆ＋１番地の値０のうち、＜ｇ番地の値ｅ＞および＜ｆ番地の値０＞という２つの番地および値の組、あるいはこのどちらか一方の組が、変更データとして他方の計算機のコピー側データ構造管理装置３０に送信される。

【００９７】この変更データを、変更受信部３１１を介して受信したコピー側のデータ構造管理装置の変更反映部３０１は、例えば＜f番地の値０＞のみが送られた場合、ｆ番地が図１２に示すように、バイナリー・ツリーのノードに使用されていることを前提にして、このｆ番地のノードをバイナリー・ツリーから削除する操作を行なう。このノードの削除の操作は、先に変更を行なったマスター側のデータ構造管理装置１０と同じ手続で行なわれる。結果として、図１４と同一である構成が変更反映部３０１によって作り出される。

【００９８】一方、＜ｇ番地の値ｅ＞のみが送られた場合でも、図１２において、図１３の標準構造をとったときにｇ番地が持つべきポインタの値ｆは、バイナリー・ツリー中の中間順序でｇの左隣のノードの番地を計算することで求めうる。この変更データにより変更反映部３０１は、このｇ番地のポインタ値がｅに変更されたことを検知し、この変更データにに対応して図１５においてｇ＋１に変更される前の図１３の標準構造中のｅ＋１番地の値がｆ＋１であることを、上記のｇ番地の場合と同様にして求める。これにより、変更前のｇ番地の値がｆ、ｅ番地の値がｆ＋１であることがそれぞれ確認され、ノードｆを削除する操作がマスター側のデータ構造管理装置１０側で行なわれたことが判定できる。この判定に基づいて、コピー側のデータ構造管理装置３０は、マスター側と同じ手続でｆ番地のノードを削除する。

【００９９】要するに、標準構造に照らしてどのような変更操作が行なわれたかを正確に判定するに足る番地および値の組が、変更送信部１１１から変更受信部３１１に送信されればよい。例えば、ｆ番地の内容を実際には０にしなくても、削除の操作に限っては、＜f番地の値０＞を送信することで、ｆ番地のノードを削除したことをコピー側に伝えることができる。

【０１００】尚、これらのａからｊの番地がノードとして使用されていることをコピー側のデータ構造管理装置が判断するためには、記憶装置上に記憶内容の一部としてこれらの番地を保持しておいてこれを参照してもよい。あるいは、操作を行なう場合にはそれと知って操作しているのだから、変更送信部１１１において送出する番地および値の組に、当該番地がノードの（偶数）番地である旨を付加して、コピー側の変更受信部３１１に送信してもよい。

【０１０１】また、対象データ構造を全く保持していなかった（コピー側）データ構造記憶部２０にマスター側データ構造全体をコピーする１つの方法として、マスター側データ構造管理装置１０上に、変更履歴の出力を全て保存しておき、この変更履歴をコピー側データ構造管理装置３０に逐次入力してデータ構造を復元してもよい。

【０１０２】上記のように、実際に変更されたポインタよりも少ないポインタの変更データを送るだけで、一方のデータ構造での変更を、他方のデータ構造に効率よく反映させることができる。

【０１０３】（２）ノードの追加の場合

次に、データ構造にノードを追加する場合の変更反映手続を説明する。

【０１０４】例えば、図１４のデータ構造において、ｆ番地のノードを再びｐ番地をポットとするバイナリー・サーチ・ツリーに追加することを考える。

【０１０５】図１６は、このｆ番地のノードを追加した後のバイナリー・サーチ・ツリーのデータ構造を示す。まず５０をキーとしてツリーを探索し、ｆ番地のノードを図１６に示すように右の子がないノード ｅの右の子として追加することができる。図１６に対応する標準構造は、図１２と同様に図１３であるから、図１３の標準構造と図１５の構造との違いは、ｇ番地の値ｆ、ｅ＋１番地の値ｆ＋１、ｆ番地の値e、ｆ＋１番地の値ｇ＋１の４つのポインタである。

【０１０６】マスター側の変更送信部１１１は、これらの４つのポインタのうち、＜ｆ番地の値ｅ＞および＜ｇ番地の値ｆ＞のいずれか一方をコピー側の変更受信部３１１に送信する。

【０１０７】＜ｆ番地の値ｅ＞を受信した場合、コピー側の変更反映部３０１は、図１４の状態においてｆ番地の値が０であることを調べる。変更反映部３０１は、送信されてきたポインタ値が、図１４に対応する図１５の標準構造において、ツリーにに属していなかったノード ｆをノード ｅの右の子として追加した図１３の標準構造に対応するいずれかの構成、例えば図１６の構成に対して図１４の構成を変更する操作を指示するものと判定し、中間順序でノード ｅの右隣（直後）になるようにノード ｆを追加する操作を実行する。この追加操作の

手順は、例えば以下の手順で行うことができる。

【０１０８】a.指定されたノードをｘとし、中間順序でその直前（または直後）になるように、追加されるノードｙを追加する。すなわち、

b.ｘに左（または右）の子がなければ、ｙをｘの左（または右）の子として処理を終了する。ｘに左（または右）の子があれば、その左（または右）の子を新たにｘとする。

【０１０９】c.ｘに右（または左）の子がなければ、ｙをｘの右（または左）の子にして処理を終了する。ｘに右（または左）の子があれば、その右（または左）の子を新たにｘとして再びｃ．を実行する。

【０１１０】d.空のツリーを指定した場合は、追加されるノードｘをルートとする。

【０１１１】一方、＜ｇ番地の値ｆ＞を受信した場合は、変更反映部３０１は、ノード ｇがツリーに属しており、かつその標準構造におけるｇ番地の値がｆでなくｅであることを判定した上で、ノード ｆを中間順序でノード ｇの左隣（ソート順で直前）になるように追加する操作を実行する。この追加操作も、上記のa.からd.の機能により行うことができる。

【０１１２】いずれの場合も、図１６に示す同一の構成を得ることができる。

【０１１３】上記のようにして、変更されたポインタのすべてを逐一送るのではなく、標準構造におけるポインタの変更の一部を送るだけで、一方に加えられた変更を他方に正確に反映してそれらの同一性（機能的な等価性）を維持することができる。尚、上記の具体例では、２つのデータ構造について説明したが、３つ以上の場合にも、直接変更を加えたデータ構造からそれ以外のデータ構造に同一のポインタ値変更データを送ることにより、複数のデータ構造間で同一性を保つことができる。

【０１１４】また、伝達されるポインタの変更データは、標準構造におけるポインタ値及びポインタ格納番地そのものでなくてもよい。変更操作に対応して、送出データ抽出部１０７で効率的に作成でき、かつ、コピー側の変更反映部３０１でも効率的に操作内容が復元できる、ポインタ変更データと等価な変更データであればどのような形式でも構わない。

【０１１５】尚、第１の実施形態においては、バイナリー・ツリーのルートへのポインタを格納した記憶要素をポットとし、ポットへのポインタをツリーのルートノードに格納している。この構成により、コピー側および／またはマスター側での変更対象ノードがルートであっても、ルート以外のノードの変更の場合と同様に、ノードに格納されたポインタの変更情報を伝達するだけで、コピー側の変更反映部３０１にマスター側データ構造における変更内容を正確にコピー側データ構造に反映させることができる。

【０１１６】一例として、＜ｆ番地の値０＞を受信したコピー側のデータ構造が図１２に示すようにノードｆをルートとする場合を説明する。[f]に格納されたポットへのポインタｐを知ることによって、ｆを削除した後の新たなルートノード、例えばノードｇの要素[g]にｐを設定し、かつポットｐの要素[p]にｇを設定して、削除後の正しいデータ構造、例えば図１４を作成することができる。

【０１１７】バイナリー・サーチ・ツリーに関しては、機能的に等価であっても、異なる多数の構造をとることができ、任意のノードをルートとする構造があり得る。マスター側とコピー側とでツリーの構造が異なる場合には、変更対象のノードが、マスター側ではルートでなくともコピー側ではルートになっているという状況があり得る。ルートノードにポットへのポインタを格納しない図３のような構成の場合には、ルートノードの変更に伴って、これを参照している（ポットに相当する）変数を指定して、新たなルートを参照するように変更する必要がある。マスター側では変更対象ノードがコピー側でルートになっているか否かを知ることはできない。このため、どのノードを変更した場合でも、変更に伴ってコピー側でルートを参照する変数の変更の可能性が少しでもあれば、これらの変数のアドレスを変更情報に付加して送出しなければならない。

【０１１８】第１の実施形態のように、ルートにポットを指示するポインタを格納すれば、必要に応じてツリーの各ノードからルートを経由してポットのアドレスを得ることができるので、ポットのアドレスを指示されなくてもルートノードの変更操作を行うことができる。従って、マスター側からコピー側にポットのアドレスを伝達する必要はない。但し、空のポットにノードを追加する場合や、標準構造におけるルートノードを変更する場合には、ポットのアドレスが送信される。しかし、これらの場合でも、ポットのアドレスはノードの要素の新たなポインタ値としてノードのアドレスと同様に扱われ、特別な付加データとして送信されるのではない。

【０１１９】尚、第１の実施形態ではツリー（のルート）を参照する変数をポット１つとして説明したが、ツリーを参照する変数が複数ある場合にも、１つのポットを経由してツリーにアクセスする図１９のような構成を採れば、上述の方法を何ら変更することなくマスター側とコピー側とでデータ構造の等価性を維持することができる。さらに、図１９に示すように、ポット側に当該ポットを参照する各変数へのポインタを格納すれば、各変数がどのポットを参照しているかという状態の変更を、これらポット側のポインタの変更として伝達することによって処理できる。

【０１２０】第１の実施形態では、ポットへのポインタを、ルートノードの、親または右の兄弟へのポインタを格納する記憶要素に格納していたので、ポットを含めたデータ構造を構成するのに、何ら余分な記憶リソースを

要することがない。この他、こうしたコンパクトな実現方法の他に、図２０に示すように、各ノードにポットへのポインタを格納する構成を採ることもできる。この場合には、ルートノードの親または右の兄弟へのポインタを格納する要素には、ポットへのポインタを格納する必要がない。

【０１２１】第１の実施形態は、バイナリー・ツリーのルートへのポインタを格納した要素をデータ構造に含め、少なくとも１つのノードにポットへのポインタを格納することによって、ポットの番地を指定しなくてもルートの変更操作を行うことができる。このため、マスター側からコピー側へ変更情報を伝達する際に、ノードに格納されたポインタの変更情報のみを送信するだけでよく、ツリー（のルート）を参照する変数の番地を付加する必要がない。また、このような構成においては、ポットは１つとは限らず、複数設けることもできる。

【０１２２】また、第１の実施形態で扱うバイナリー・ツリーに限らず、一般にポインタによって構成される動的なデータ構造は、その一部の記憶要素のアドレスを知って、その要素から探索等のアクセスが開始される。バイナリー・ツリーではルートノードがこのような要素であり、そこからバイナリー・サーチが適用される。動的に変更されるデータ構造では、このような要素（例えばルートノード）は変更されうるが、それに伴って、このような要素を参照している要素（変数）も変更する必要がある。これらの変数を変更するためには、これらの変数の番地あるいは番地を決定するための情報をマスター側からコピー側に伝達しなければならない。

【０１２３】第１の実施形態は、データ構造の一部として動的に番地が変わらない記憶要素（ポット）を設け、動的に番地が変更されうる要素（ルートノードあるいは各ノード）からこの番地が変わらない要素への対応をデータ構造の機能として記憶する、即ち、ルートノードあるいは各ノードにポットへのポインタを格納する。このため、変更操作のたびにデータ構造への参照を保持する変数の番地をマスター側からコピー側へ伝達する手間を省くことができる。

【０１２４】さらに、マスター側からコピー側への伝達の手間が減少するだけでなく、単独のデータ構造（例えば、マスター側のみ）の変更操作を行う際にも、データ構造変更部への入力としてポットの番地（データ構造への参照を保持する変数の番地）を指定する必要がない。これにより、変更操作の前提としてデータ構造への参照を保持する変数の番地を知らなければならないという制約が除去され、データ構造への変更操作の適用の自由度が著しく向上する。

【０１２５】また、例えば、データ構造からノードを削除する操作を行う際に、削除対象ノードがそのデータ構造の唯一のノードである場合には、そのデータ構造すなわちそのノードを参照している変数ポットを、無効なノードを参照させないように変更する必要がある。このためには、ポットの番地を知る必要があるが、ノードから当該ノードが属するバイナリー・ツリーのポットを知る手段がなければ、削除対象ノードとはべつにポットを示す情報を指定しなければならない。この指定は、上記のように全ての削除操作において必要なわけではないが、

（１）必要か否かの判定を行ったうえで指定する。

【０１２６】（２）必要か否かに拘わらず指定する。

【０１２７】（３）必要な状況に陥らないようにデータ構造の使用を制限する。例えば、常に一つは（ダミーであっても）ノードを有するように使用する。

【０１２８】等のいずれの方法をとっても、余計な時間や記憶リソースを要する。特に指定すべきポットを知り得ない状況で削除操作が行われるおそれがある場合は、このための回避策は上記の（３）しかない。

【０１２９】同様の状況は、ルートノードが削除される（削除すべきノードがたまたまルートノードであった）場合や、スプレイング操作等でルートノードが交替する場合にも生ずるため、上記の（１）、（２）または（３）のような対策に余計なリソースを要する機会は少なくない。

【０１３０】第１の実施形態のように、ノードからポットを知る手段がデータ構造に備わっていれば、真に必要な場合以外はポットを指定する必要がなく、操作対象ノードのみを指定して変更操作を行うことができるだけでなく、上記（３）のようにデータ構造の使用を制限したり、余分なリソースを要することもない。

【０１３１】第１の実施形態によれば、以下の効果が得られる。

【０１３２】第１の実施形態に係るデータ構造管理システムは、任意のデータ構造の標準構造におけるポインタ値の変更に着目して、変更されたポインタ値のうち、標準構成を前提にすれば変更反映部３０１側で推定および復元可能なポインタ値を省略して、変更データを生成し、伝達する。これにより、データ構造を変更するプログラムに意識させることなく、変更データの通信量を減少させるとともに、変更反映処理の処理時間を削減することができる。

【０１３３】尚、第１の実施形態の変形例として、上記の追加の例において、コピー側のデータ構造が図１７に示すように、図１４のマスター側データ構造と異なる構成である場合にも、上記の変更反映処理を適用することができる。

【０１３４】＜ｆ番地の値ｅ＞または＜ｇ番地の値ｆ＞を受信して、上記の機能を使用してノードｆの追加を行なうことによって、図１６とバイナリー・サーチ・ツリーとしては等価な図１８の構成に変更することができる。

【０１３５】このように、マスター側とコピー側が異なる構成であっても、あるいは複数のコピーがそれぞれ異

なる構成であっても、標準構造におけるポインタの変更の一部を伝送するだけで、それらの間の等価性を維持することができる。

【０１３６】第２の実施形態以下、図２１から図２３を参照して、本発明に係るデータ構造管理装置、データ構造管理システム、データ構造管理方法およびデータ構造管理プログラムの第２の実施形態を、第１の実施形態と異なる点についてのみ、詳細に説明する。

【０１３７】第２の実施形態は、第１の実施形態が行うコピー側データ構造へのポインタ情報であるアドレスの送出に替えて、相対的位置情報を送出する機能を提供する。具体的には、第２の実施形態は、データ構造の各ノードに、キーの他に「区間管理方法」（特許第２７６８９２１号）に開示されている部分区間長を格納し、区間の位置によるノードの探索およびノードに対応づけられた区間の算出を行なう。

【０１３８】図２１は、本発明の第２の実施形態に係るデータ構造管理装置の機能構成を示すブロック図である。図２１に示すように、第２の実施形態は、図１の第１の実施形態と比較して、マスター側データ構造管理装置１０中に位置探索部１１３と、コピー側データ構造管理装置３０中に位置探索部３１３とをさらに具備する点において、第１の実施形態の変形である。

【０１３９】位置探索部１１３および位置探索部３１３は、区間の位置によるノードの探索およびノードに対応づけられた区間の算出を行なう。

【０１４０】その他の構成は、図１の第１の実施形態と同様であるため、説明は省略する。

【０１４１】図２２（ａ），（ｂ）は、第２の実施形態におけるデータ構造の一例を示す。図２２（ａ），（ｂ）において、各ノードに対応づけられた区間の長さは全て１であり、各区間の端はそのノードのＬＣＲオーダーにおける左からの順位を表わしている。

【０１４２】マスター側のデータ構造管理装置１０の送出データ抽出部１０７ｂは、標準構造記憶部１０９中の標準構造において変更操作の結果変更されたポインタ値及びポインタの格納番地を、位置探索部１１３を介して位置情報として得る。この位置情報は、変更以前の状態におけるその番地を含むノードの先頭からその番地までのオフセットと、そのノードが所属するバイナリー・ツリー中でのそのノードの区間（順位）の値と、そのバイナリー・ツリーのポットの番地との組で表わされる。この位置情報は、コピー側のデータ構造管理装置３０の変更受信部３１１に送出される。

【０１４３】コピー側のデータ構造管理装置３０の変更反映部３０１は、変更受信部３１１により受信された相対位置情報である変更データに基づいて、ポットの番地からそのポットに対応するツリーのルートを決定し、区間（順位）を指定して位置探索部３１３に指示する。位置探索部３１３は、その位置を含む区間（その順位）の、コピー側データ構造におけるノードの番地（先頭番地）を決定し、決定された先頭番地にオフセットを加えることにより、コピー側のデータ構造における対応する番地を算出する。

【０１４４】図２２（ａ），（ｂ）においては、例えば番地ｆは「ｑ番地のポットの順位１のノードのオフセット０の番地」と表わされる。また番地ｅは「ｐ番地のポットの順位５のノードのオフセット０の番地」と表わされる。従って、変更送信部１１１が送出する変更データの内容としては、例えば＜ｆ番地の値ｅ＞の替わりに、＜ｑ番地のポットの順位１のノードのオフセット０の番地に、ｐ番地のポットの順位５のノードのオフセット０の番地が格納された＞という情報を示す位置情報となる。

【０１４５】尚、データ構造の状態および変更操作に関して、第１の実施形態と異なる点は、ポット以外の番地はノードからのオフセットにより指定される点である。このために、ノードｆはポットｐのツリーに追加される前にも、ポットｑのツリーに属していることとなる。変更操作としては、図２３（ａ），（ｂ）に示すように、いったんノードｆを削除してから、これをＬＣＲオーダーでノードｅの右隣の位置になるように追加する。

【０１４６】第２の実施形態によれば、以下の効果が得られる。

【０１４７】すなわち、相対的位置情報を変更データとして送出するので、マスター側のデータ構造とコピー側のデータ構造とで、対応するノードが必ずしも同一の番地を占める必要がない。このため、マスター側とコピー側とで記憶領域（アドレス空間）の大きさや使い方が異なっていても、上記のデータ構造の等価性を実現でき、本発明の適用対象を著しく拡大することができる。

【０１４８】尚、バイナリー・ツリー（バイナリー・サーチ・ツリー）において、ツリー中のノードを特定する手法として、上記の区間管理方法による以外の手法を用いることができる。例えば、図２３（ａ），（ｂ）に示すように、各ノードのキーが全て異なる場合のように、変更の対象となるノードと同じキーを持つノードが変更前後のツリー中に存在しなければ、当該ノードの特定のために、キーを用いてもよい。

【０１４９】また、上記の説明は、複数のデータ構造間でポットの番地が一致している場合に、ノードの番地が異なることを許すものであるが、逆にノードの番地が共通で、ポットの番地が異なることを許すようにもできる。但し、標準構造における変更に関しては、ポットの番地が変更データとして送られるのはルートである左端のノードが変わる場合だけである。

【０１５０】尚、上記の方法の拡張として、データ構造ごとに番地が異なってもよいノードおよびポットから成る部分に関しては、そのノードまたはポットを特定するデータ及びオフセットの組で共通の番地からそのデータ

構造における対応する番地を算出するに足るデータを、マスター側の変更送信部１１１ｂとコピー側の変更受信部３１１との間で送受するという手法が可能である。

【０１５１】第３の実施形態

以下、図２４から図２６を参照して、本発明に係るデータ構造管理装置、データ構造管理システム、データ構造管理方法およびデータ構造管理プログラムの第３の実施形態を、第１および第２の実施形態と異なる点についてのみ、詳細に説明する。

【０１５２】図２４は、第３の実施形態に係るデータ構造管理装置の機能構成を示すブロック図である。図２４に示すように、第３の実施形態は、第１および第２の実施形態と異なり、マスター側に標準構造記憶部１０９を持たず、図１のデータ構造変更部１０５と送出データ抽出部１０７の機能を併せ持つデータ構造変更・抽出部１０６を具備する点において、第１の実施形態の変形である。

【０１５３】第１および第２の実施形態においては、データ構造変更部１０５から出力されるポインタ変更情報のうち、標準構造記憶部１０９に記憶されている標準構造における変更のみを送出データ抽出部１０７、１０７ｂが抽出あるいはさらに適宜変換して変更送信部１１１、１１１ｂに出力する。一方、第３の実施形態においては、データ構造記憶部２０のデータ構造をデータ構造変更・抽出部１０６が変更する際に、変更対象のデータ構造上で変更対象ポインタの標準構造における値を算出し、必要な変更データのみを変更送信部１１１に出力する。

【０１５４】これにより、標準構造記憶部１０９に標準構造を保持するのに要する資源およびデータ構造の変更に伴ってこの標準構造を変更するための手間を省くことができる。同時に、データ構造変更部１０５から一旦出力された変更データを、標準構造に照らして抽出するという送出データ抽出部１０７の手間を軽減することができる。

【０１５５】第３の実施形態は、データ構造として、第１の実施形態と同じバイナリー・ツリーを使用する。データ構造変更・抽出部１０６は、変更操作に応じて、必要な変更データを抽出するためにデータ構造記憶部２０に記憶されている変更対象のデータ構造そのものを使用する。

【０１５６】以下、第３の実施形態におけるデータ構造変更に伴う変更反映処理の処理手順を説明する。

【０１５７】（１）ノードの削除の場合

例えば、図１２のデータ構造においてノードｆを削除する際に、変更データとして＜ｇ番地の値ｅ＞または＜ｆ番地の値０＞を得る必要がある。＜ｆ番地の値０＞は特に標準構造に照らさなくても作成できるが、＜ｇ番地の値ｅ＞は図１２のデータ構造からは容易には得られない。第３の実施形態におけるデータ構造変更・抽出部１０６は、self-adjusting binary treeの操作であるスプレイング（splaying）を使用して、図１２のデータ構造を変更、すなわちノードｆを削除するとともに、削除後の標準構造における＜ｇ番地の値ｅ＞を以下のように決定する。

【０１５８】(i)まず、削除対象のノードｆにスプレイングを適用してｆをルートノードとする。図１２では既にｆがルートノードになっているので、この操作が終わった状態と考えればよい。

【０１５９】(ii)次に、ＬＣＲオーダーでノードｆの右隣のノードｇおよび左隣のノードｅを求め、ｇにスプレイングを適用して、図２５に示すように、ｇをルートノードに移動する。

【０１６０】(iii)最後にｆの左隣のノードｅにスプレイングを適用してｅをルートに移動すると、ノードｅ，ｆおよびｇの間の構造は、図２６に示すようになる。

【０１６１】(iv)この図２６の状態でノードｆを削除すれば、ノードｇはＬＣＲオーダーで左隣のノードｅの右の子となり、ノードｇ自身は左の子を持たない構造となる。この構造はノードｆ削除後の標準構造図である図１５におけるｅとｇ　との関係と一致しており、＜ｇ番地の値ｅ＞も同一である。

【０１６２】このように、ノードを削除する場合、削除対象ノード、右隣のノード、左隣のノードの順にスプレイングを行った後に削除を行えば、左隣のノードと右隣のノードとの間の構造は必ず、「右隣のノードは左の子を持たず、かつ左隣のノードの右の子になっている」という標準構造に一致する。これを利用して、変更送信部１１１は、データ構造記憶部２０のデータから直接送信すべき変更データ（上記の例では＜ｇ番地の値ｅ＞）を抽出することができる。

【０１６３】（２）ノードの追加の場合

図１７のデータ構造において、ノードｆを追加する場合は、以下の手順を実行する。

【０１６４】(i)まず、ノードｆを適切な位置に追加して、例えば図１８のようなデータ構造に変更する。

【０１６５】(ii)次に、ノードｆにスプレイングを適用してルートに移動し、さらにノードｆの左隣のノードｅを求めてノードｅにスプレイングを適用する。

【０１６６】(iii)その結果、ノードｆは左の子を持たず、かつ（ルートノード）ｅの右の子となって、ノードｅとノードｆとの関係は図１３の標準構造に一致する。得られたこの構造から、変更送信部１１１が送信すべき＜ｆ番地の値ｅ＞を抽出することができる。

【０１６７】(iv)同様に、ＬＣＲオーダーでノードｆの右隣のノードを求め、まずノードｇにスプレイングを適用した後に、続けてノードｆにスプレイングを適用する。

【０１６８】(v)その結果、ノードｇは左の子を持たず、かつ（ルートノード）ｆの右の子となって、ノード

ｆとノードｇとの関係は、図１３に示す標準構造に一致する。この構造から、変更送信部１１１が送信すべき＜ｇ番地の値ｆ＞を抽出することができる。

【０１６９】以上の手順により、特に記憶された標準構造を参照しなくても、データ構造記憶部２０のデータ構造を使用して標準構造におけるポインタ値を計算して変更送信部１１１に出力するようにデータ構造変更・抽出部１０６を構成することができる。

【０１７０】尚、第３の実施形態においては、スプレイングを適用して実際にデータ構造を変形した後に、所定のノードの番地内容を読みとることによって変更データの抽出が行われた。しかし、実際にこの手続きを全て行わなくても、操作の種類（ノードの削除、追加）ごとに、操作対象ノードとＬＣＲオーダーとによりその左隣または右隣のノードが求められれば、最終的に抽出される変更データを作成することができる。この場合には、標準構造はデータ構造記憶部２０のデータ構造に部分的にも保持されている必要はない。ここで抽出される変更データは、指定された標準構造におけるポインタの値であるため、データ構造変更・抽出部１０６は、標準構造を参照するのと等価な計算を行っていると言いうる。

【０１７１】本発明の１つの特徴は、変更送信部１１１から変更受信部３１１に伝達される変更データが標準構造におけるポインタ（変数の）値であることである。この変更データがいかなる方法によって抽出、計算されていてもその機能および効果に差異はない。従って、抽出、計算の手法に種々の方法を適用しても本発明の本旨を逸脱することはない。

【０１７２】尚、上記の実施形態では、データ構造（二分探索木）としてソートされたデータを扱う。この標準構造は、データのソート順とノードの結合順が対応するように設定されており、ソート順に影響を及ぼす変更操作のみがノードの結合順を変えるように構成されている。同一のソート順のデータを表していれば、データ構造が異なっていても機能的な差異はない。従って、上記の実施形態は、ソートされたデータを扱うデータ構造に関して、機能的な差異を生ずる変更のみを標準構造におけるポインタの変更として抽出するものの一例である。

【０１７３】尚、本発明は上述した実施形態に限定されるものではなく、本発明の本旨を逸脱することなく、種々変更・変形を成し得ることが可能であることは言うまでもない。また、これらの変更・変形はすべて本発明の範囲内に含まれるものである。

【０１７４】

【発明の効果】以上説明したように、本発明によれば、以下の効果を奏する。

【０１７５】すなわち、本発明に係るデータ構造管理装置、データ構造管理システム、データ構造管理方法およびデータ構造管理プログラムを格納する記録媒体は、１つまたは複数の計算機上の、１つまたは複数の記憶領域上の複数のデータ構造の等価性を、わずかな通信量の変更データを送受することによって容易に効率よく維持することを可能とする。また、それぞれ異なる番地に格納されるデータ構造であっても、これらの等価性維持を可能とする。また、ネットワーク上の複数の計算機間でポインタ構造を効率よく複製することができ、例えば高信頼性システムなど多様なシステムに適用できる。

【０１７６】このように、本発明を用いれば、データ構造をネットワーク上の複数の計算機に共有させ、これらデータ構造を共有する各計算機上で記憶領域の管理をそれぞれ独自に効率よく行うことができる。

【図面の簡単な説明】

【図１】本発明の第１の実施形態に係るデータ構造管理システムの機能構成を示すブロック図である。

【図２】本発明の第１の実施形態に係るデータ構造管理方法の概略の処理手順を示すフローチャートである。

【図３】２ポインタ表現のバイナリー・ツリーのポインタ構造を説明する図である。

【図４】図３のバイナリー・ツリーのツリー構造を説明する図である。

【図５】バイナリー・ツリーのノードに格納されるポインタ（番地）を説明する図である。

【図６】ツリーが空の場合のポットのノードのポインタを説明する図である。

【図７】ポットとルートとのノードのポインタ関係を説明する図である。

【図８】子がないノードのポインタを説明する図である。

【図９】右の子のみを持つノードのポインタを説明する図である。

【図１０】左の子のみを持つノードのポインタを説明する図である。

【図１１】左右両方の子を持つノードのポインタを説明する図である。

【図１２】図３のバイナリー・ツリーを、ポットを含めて記述したデータ構造の一例を示す図である。

【図１３】図１２のデータ構造と等価である標準構造の一例を示す図である。

【図１４】変更操作を行った後の図１２のデータ構造の一例を示す図である。

【図１５】図１４と同様の変更操作を行った後の図１３の標準構造の一例を示す図である。

【図１６】図１４のデータ構造に追加の変更操作を行った後のデータ構造の一例を示す図である。

【図１７】マスター側のデータ構造と異なるコピー側のデータ構造の一例を示す図である。

【図１８】図１７のデータ構造に対するノードの追加操作で得られる、図１６のデータ構造と等価であるデータ構造の一例を示す図である。

【図１９】第１の実施形態の変形例に係るポットとこの

ポットを参照する変数とのポインタ構造の一例を説明する図である。

【図２０】第１の実施形態の変形例に係るポットとデータ構造を構成する各ノードとのポインタ構造の一例を説明する図である。

【図２１】本発明の第２の実施形態に係るデータ構造管理システムの機能構成を示すブロック図である。

【図２２】第２の実施形態に係るデータ構造の一例を示す図である。

【図２３】図２２のデータ構造に対してノードｆを追加して得られるデータ構造の一例を示す図である。

【図２４】本発明の第３の実施形態に係るデータ構造管理システムの機能構成を示すブロック図である。

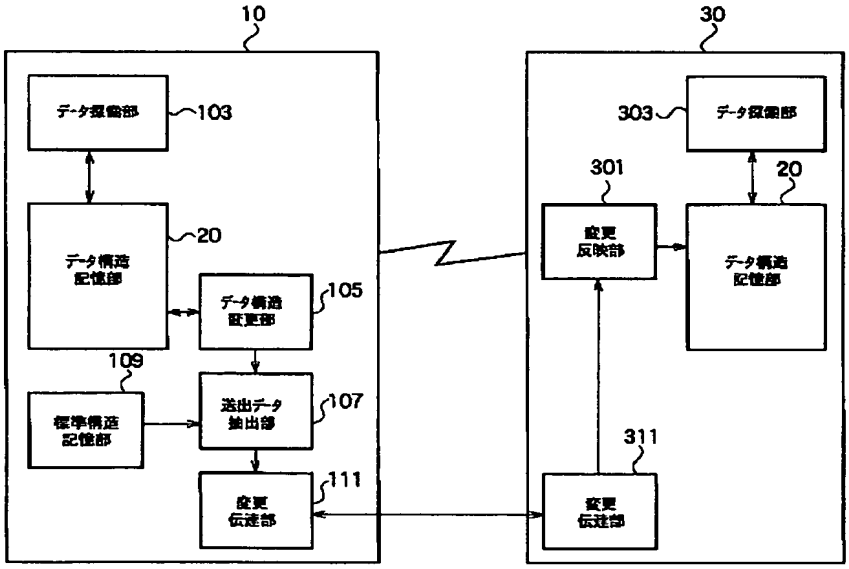
【図２５】図１２のデータ構造に対して、ルートノードｆの右隣のノードｇにスプレイングを適用してノードｇをルートに移動した後のデータ構造を示す図である。

【図２６】図２５のデータ構造に対して、ノードｆの左隣のノードｅにスプレイングを適用してノードｅをルートに移動した後のデータ構造を示す図である。

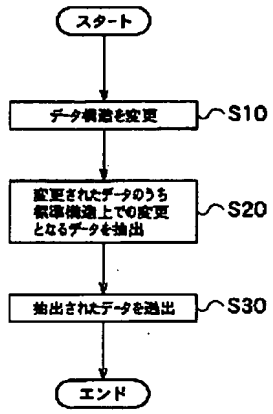
【符号の説明】

１０　データ構造管理装置
２０　データ構造記憶部
３０　データ構造管理装置
１０３　データ探索部
１０５　データ構造変更部
１０６　データ構造変更・抽出部
１０７、１０７ｂ　送出データ抽出部
１０９　標準構成記憶部
１１１、１１１ｂ　変更送信部
３０１　変更反映部
３０３　データ探索部
３１１　変更受信部
３１３　位置探索部

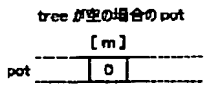
【図１】



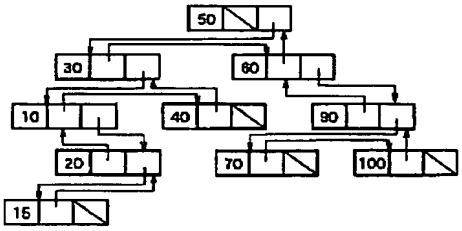
【図２】



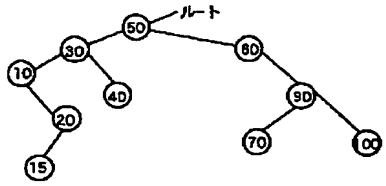
【図６】



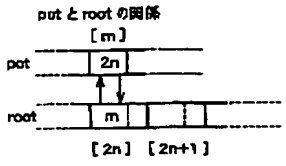
【図３】



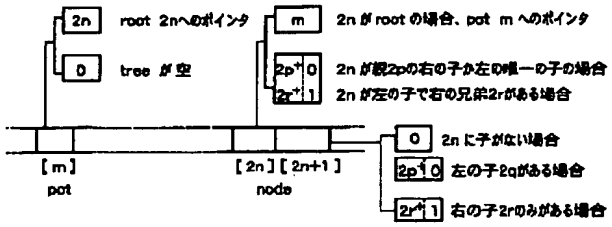
【図４】



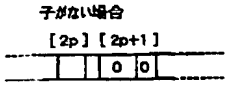
【図７】



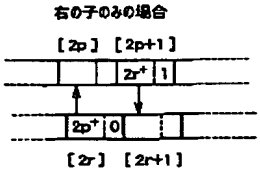
【図５】



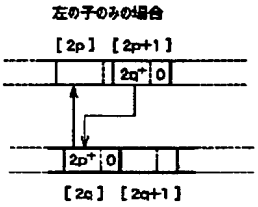
【図８】



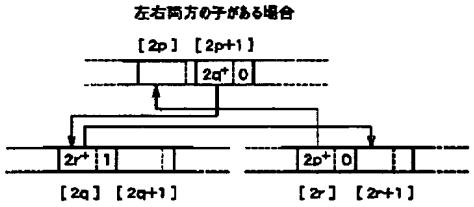
【図９】



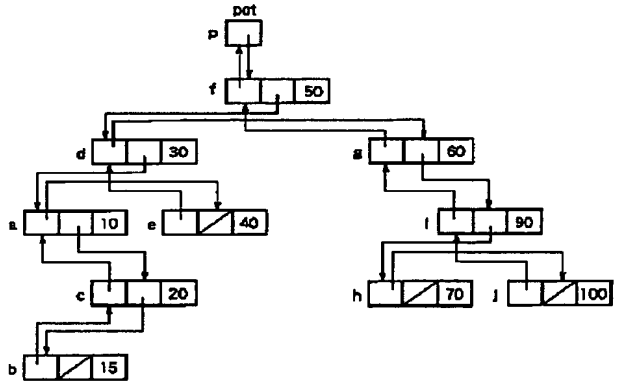
【図１０】



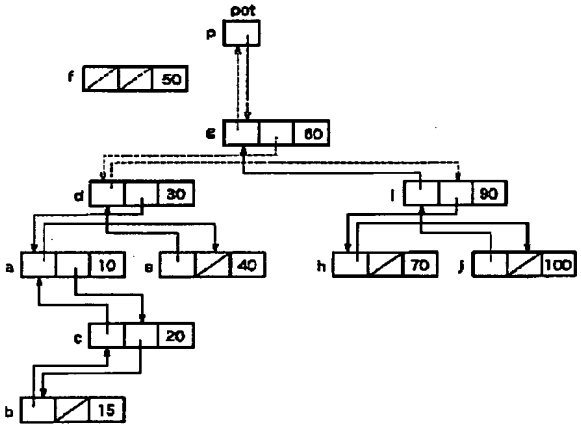
【図１１】



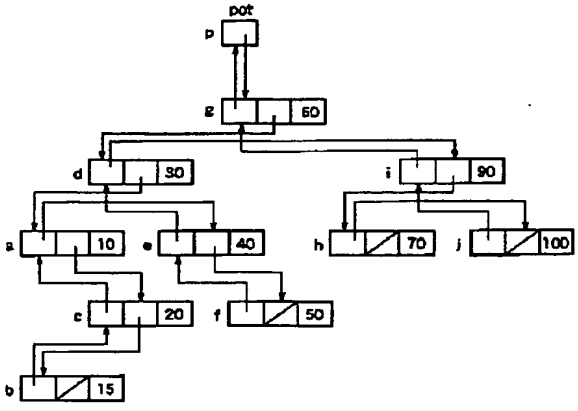
【図１２】



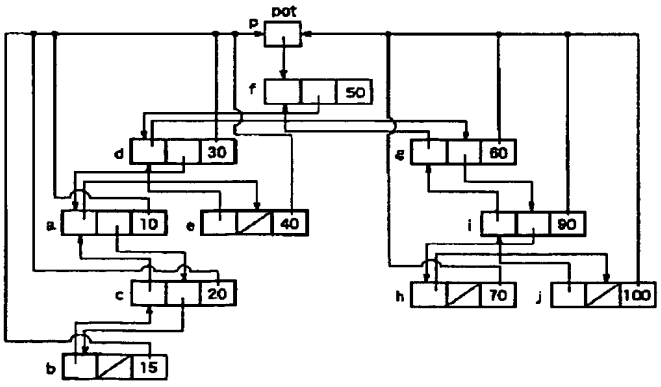
【図１４】



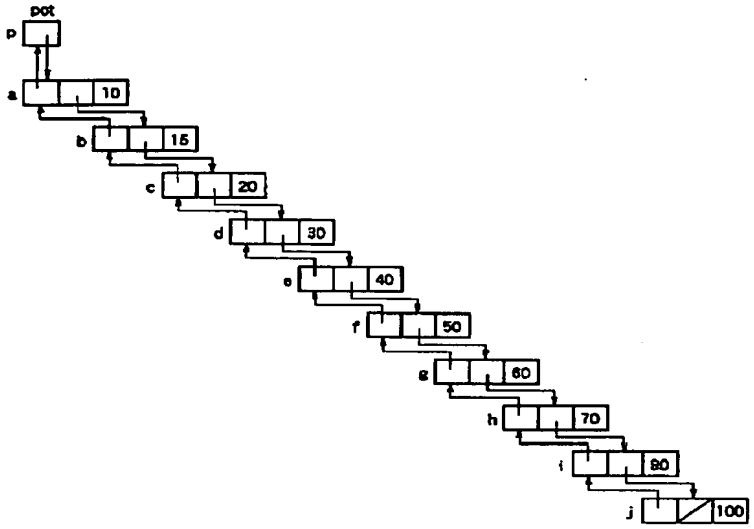
【図１６】



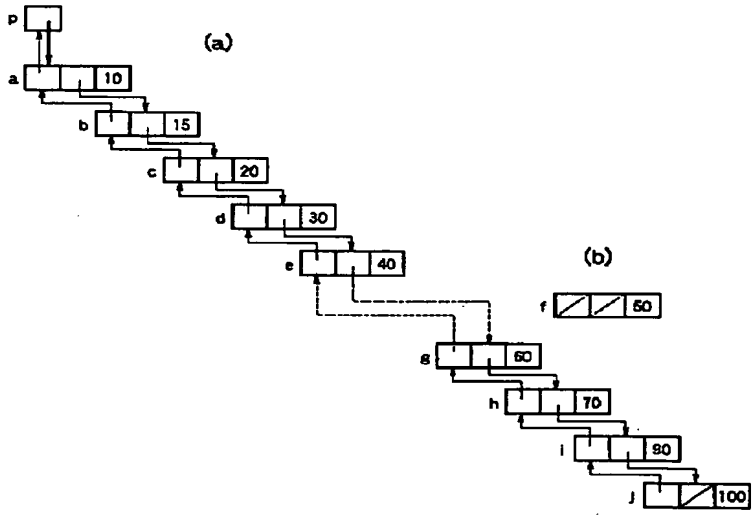
【図２０】



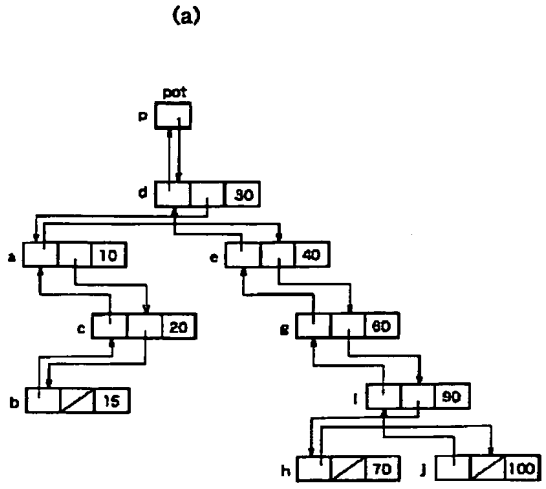
【図１３】


dot
p
a
10
b
15
c
20
d
30
e
40
f
50
g
60
h
70
i
80
j
100

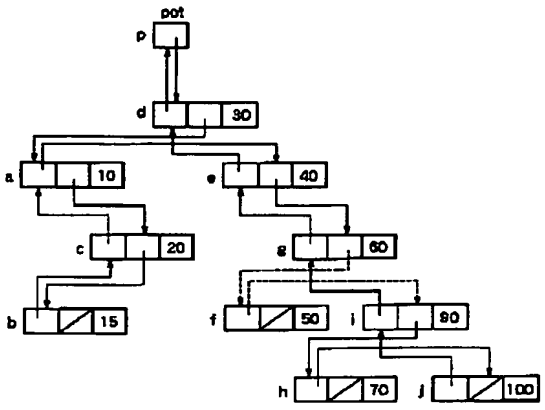
【図１５】


p
(a)
a
10
b
15
c
20
d
30
e
40
(b)
f
50
g
60
h
70
i
80
j
100

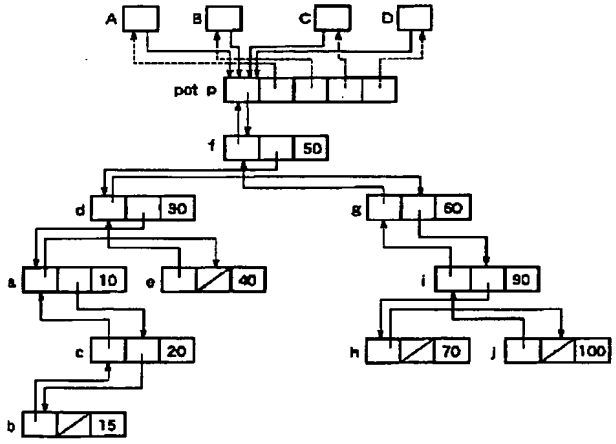
【図１７】



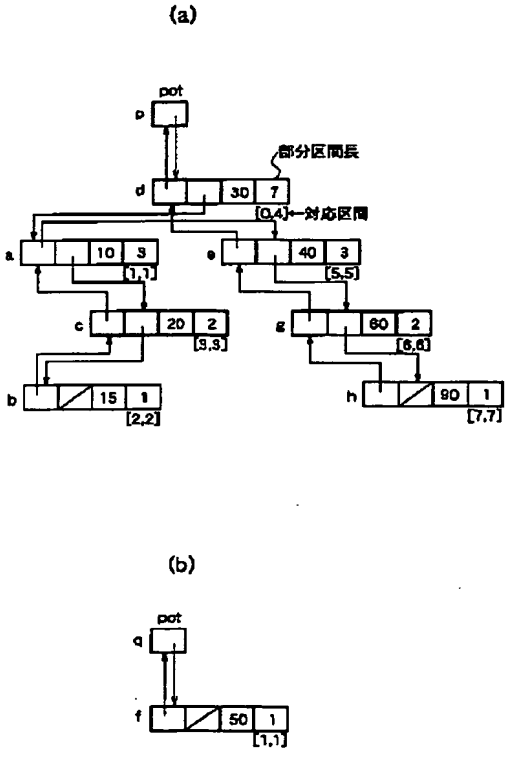
【図１８】



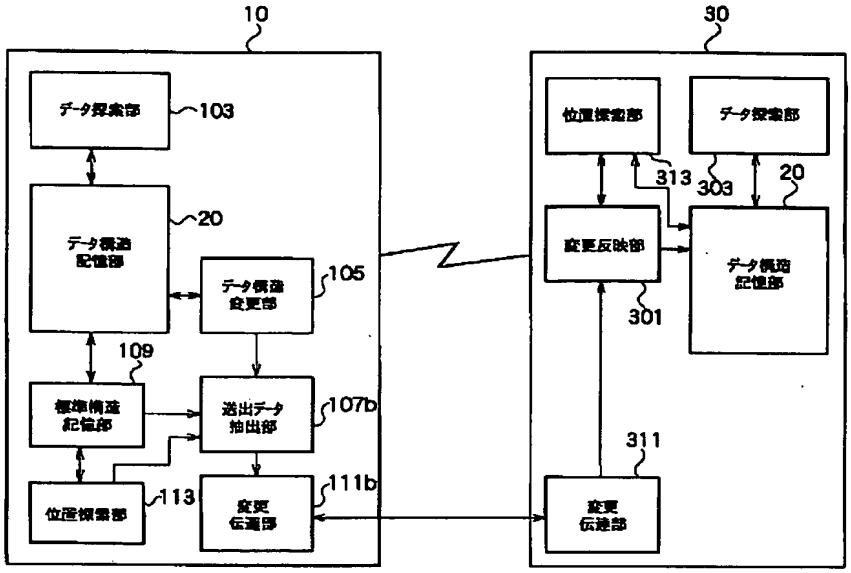
【図１９】



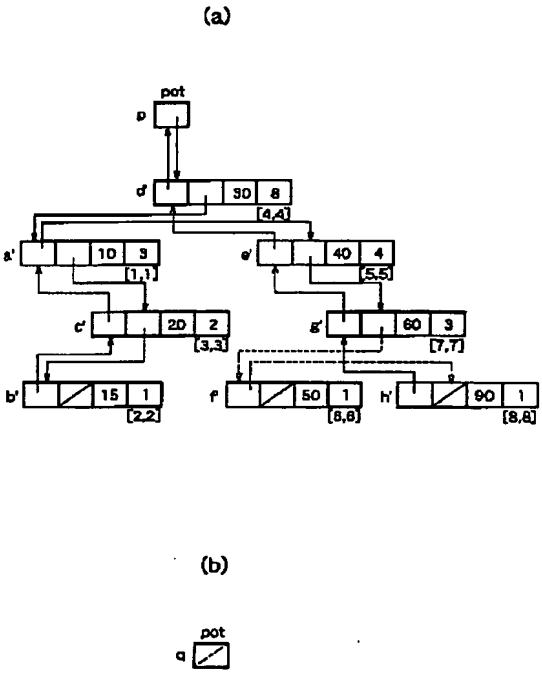
【図２２】



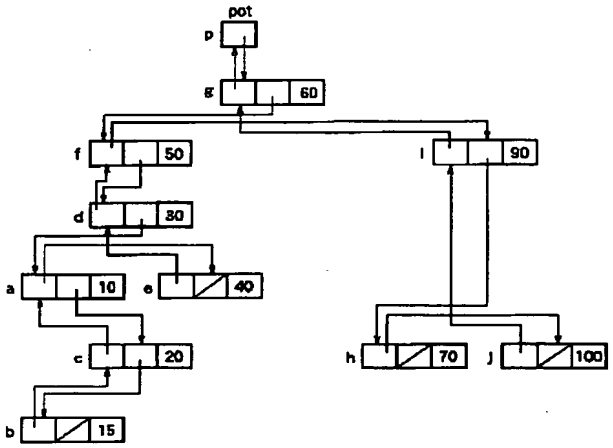
【図２１】


10
30
データ探索部
103
20
データ構造記憶部
データ構造変更部
105
109
標準構造記憶部
送出データ抽出部
107b
位置探索部
113
変更伝達部
111b
位置探索部
データ探索部
313
303
20
変更反映部
データ構造記憶部
301
311
変更伝達部

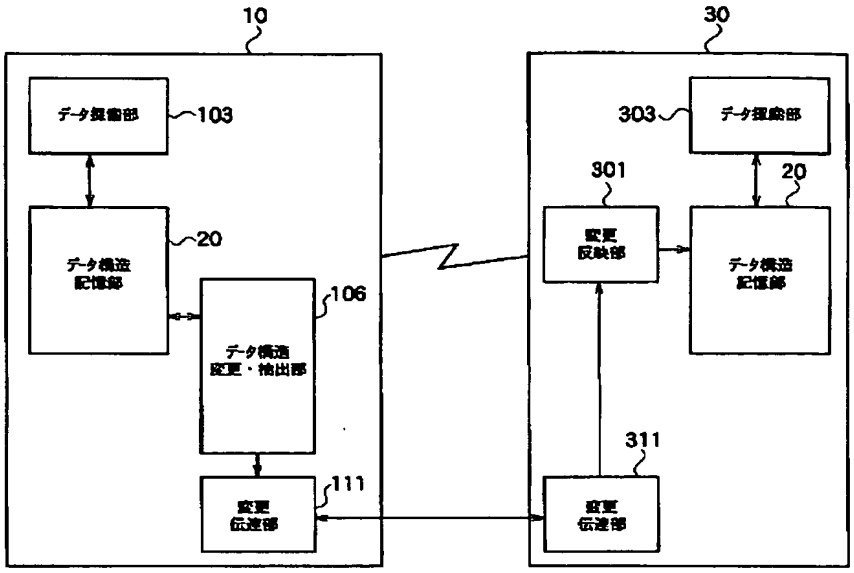
【図２３】


(a)
pot
p
d'
30
8
[4,4]
a'
10
3
[1,1]
e'
40
4
[5,5]
c'
20
2
[3,3]
g'
60
3
[7,7]
b'
15
1
[2,2]
f'
50
1
[6,6]
h'
90
1
[8,8]
(b)
pot
q

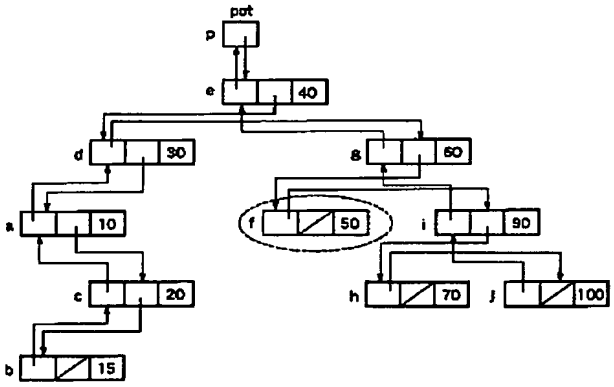
【図２５】


pot
p
g
60
f
50
i
90
d
30
a
10
e
40
c
20
h
70
j
100
b
15

【図２４】


10
30
データ操作部
103
303
データ操作部
301
20
変更
反映部
データ構造
記憶部
20
データ構造
記憶部
106
データ構造
変更・抽出部
311
111
変更
伝達部
変更
伝達部

【図２６】


pot
p
e
40
d
30
g
60
a
10
f
50
i
90
c
20
h
70
j
100
b
15